

RECEIVED
CENTRAL FAX CENTER

IN THE CLAIMS

APR 30 2007

Please amend the claims to the following.

1 1. (Currently Amended) An apparatus comprising:

2 a processor including:

3 marking logic to mark instruction information for an instruction of a speculative
4 thread as speculative; and

5 blocker logic to prevent data associated with a store instruction of the speculative
6 thread from being forwarded to an instruction of a non-speculative thread,
7 the blocker logic further to prevent the data from being stored in a memory
8 system.

1 2. (Original) The apparatus of claim 1, wherein: blocker logic is further to allow the data
2 associated with a store instruction of the speculative thread to be forwarded to an
3 instruction of a second speculative thread.

1 3. (Original) The apparatus of claim 1, further comprising: a plurality of store request
2 buffers, each store request buffer including a speculation identifier field.

1 4. (Original) The apparatus of claim 1, wherein the memory system further comprises: a
2 data cache that includes a safe-store indicator field associated with each entry of a
3 tag array.

1 5. (Cancelled)

1 6. (Original) The apparatus of claim 1, wherein blocker logic further includes:
2 dependence blocker logic to prevent data associated with a speculative store instruction
3 from being forwarded to an instruction of the non-speculative thread ; and
4 store blocker logic to prevent the data from being stored in a memory system.

1 7. (Currently Amended) The apparatus of claim 6, wherein: store blocker logic is outside the
2 an execution pipeline.

1 8. (Cancelled)

1 9. (Original) The apparatus of claim 6, wherein: dependence blocker logic is included in an
2 execution pipeline.

1 10. (Currently Amended) The apparatus of claim 9, wherein: dependence blocker logic is
2 included in a memory ordering buffer of the processor.

1 11. (Currently Amended) A system, comprising:

2 a memory system that includes a memory device dynamic random access memory; and
3 a processor associated with the memory system, the processor including dependence
4 blocker logic to prevent data associated with a store instruction of a speculative
5 thread from being forwarded to an instruction of a non-speculative thread;
6 ~~the processor further including store blocker logic to prevent the data from being stored in~~
7 ~~the memory system.~~

1 12. (Currently Amended) The system of claim 11, wherein: the processor further includes
2 store blocker logic to prevent the data from being stored in the memory system
3 and marking logic to mark instruction information associated with the store
4 instruction as speculative.

1 13. (Original) The system of claim 12, wherein: the marking logic is further to associate a
2 safe speculation domain ID with the instruction information.

1 14. (Original) The system of claim 13, wherein: the marking logic is further to indicate a
2 thread identifier as the speculation domain ID.

1 15. (Original) The system of claim 12, further comprising: a store request buffer to store the
2 speculation domain ID.

1 16. (Original) The system of claim 11, wherein: the processor includes a first logical
2 processor to execute the non-speculative thread; and the processor includes a
3 second logical processor to execute the speculative thread.

1 17. (Original) The system of claim 11, further comprising:
2 a second processor that includes said dependence blocker logic and said store blocker
3 logic;
4 wherein said processor is to execute the non-speculative thread and said second processor
5 is to execute the speculative thread.

1 18. (Currently Amended) The system of claim of claim 11, wherein: the memory system
2 includes store blocker logic to prevent the data from being stored in the memory
3 system and a cache organized to include a plurality of tag lines, wherein each tag
4 line of the cache includes a unique helper thread ID field.

1 19. (Original) The system of claim 11, wherein: the memory system includes a cache
2 organized to include a plurality of tag lines, wherein each tag line of the cache
3 includes a safe-store indicator field.

1 20. (Original) The system of claim 11, wherein: the memory system includes a victim tag
2 cache to indicate evicted cache lines that include speculative load data.

1 21. (Original) A method, comprising:

2 receiving instruction information for a load instruction, the instruction information
3 including a load address;

4 performing a dependence check, wherein performing the dependence check includes:

5 determining if a store address of an in-flight store instruction matches the load
6 address; and

7 determining if the load instruction and the in-flight store instruction each originate
8 with a speculative thread;

9 forwarding, if the dependence check is successful, store data associated with the in-flight
0 store instruction to the load instruction; and

22. (Original) The method of claim 21, wherein performing the dependence check further comprises: determining if the in-flight store instruction and the load instruction originate from the same thread.

1 23. (Original) The method of claim 22, wherein determining if the in-flight store instruction
2 and the load instruction originate from the same thread further comprises: determining if a
3 thread ID associated with the in-flight store instruction matches a thread ID associated with
4 the load instruction.

1 24. (Original) The method of claim 21, wherein performing the dependence check further
2 comprises: if the load instruction and the in-flight store instruction do not each
3 originate with a speculative thread, determining if the load instruction and the in-
4 flight store instruction each originate with a non-speculative thread,

1 25. (Original) The method of claim 21, further wherein: declining to forward further
2 comprises declining to forward the store data to the load instruction if (the load
3 instruction and the in-flight store instruction each originate with a speculative
4 thread) AND (the in-flight store instruction originates with a speculative thread
5 that is not older in program order than the speculative thread from which the load
6 instruction originates).

1 26. (Currently Amended) A method, comprising:

2 ~~processing a speculative thread cache read request;~~

3 in response to determining a cache line in a cache corresponding to a speculative thread

4 cache write request includes dirty non-speculative data;

5 generating a writeback of the dirty non-speculative data;

6 generating a writeback of the dirty non-speculative data; and

7 marking the cache line as speculative.

8 ~~processing a speculative thread cache write request; and~~

9 ~~processing a cache access request from a non-speculative thread.~~

1 27. (Currently Amended) The method of claim 26, ~~wherein processing a speculative thread~~
2 ~~cache read request further comprising~~ comprises: forwarding speculative data from a cache to
3 a the speculative thread responsive to a data speculative thread cache read request.

1 28. (Currently Amended) The method of claim 26, ~~wherein processing a speculative thread~~
2 ~~cache read request further comprising~~ comprises: forwarding non-speculative store data from
3 a cache to a the speculative thread responsive to a data speculative thread cache read request.

1 29. (Currently Amended) The method of claim 26, ~~wherein processing a cache access request~~
2 ~~from a non-speculative thread~~ further comprising comprises: forwarding non-speculative data
3 from a cache to a non-speculative thread responsive to a ~~data~~ non-speculative thread cache
4 read request.

1 30. (Currently Amended) The method of claim 26, further comprising processing a cache
2 access request from a non-speculative thread, wherein processing a cache access
3 request from a non-speculative thread ~~further~~ comprises:
4 if a cache does not include a cache line associated with the cache access request,
5 allocating a new cache line;
6 wherein allocating a new cache line further comprises:
7 if the new cache line includes dirty speculative data, allocating the new cache line
8 without generating a writeback operation; and
9 if the new cache line includes dirty non-speculative data, generating a writeback
10 operation.

1 31. (Currently Amended) The method of claim 26, ~~wherein processing a speculative thread~~
2 ~~cache write request~~ further comprising comprises: allowing the speculative thread to write
3 data to the cache if ~~a~~ the cache line corresponding to the speculative thread cache write
4 request includes speculative data.

1 32. (Cancelled)

1 33. (Currently Amended) The method of claim 26, ~~wherein processing a speculative thread~~
2 ~~each write request further comprising comprises:~~

3 if the cache does not contain data in a cache line corresponding to the speculative thread
4 cache write request data cache address:
5 allocating a new cache line;
6 marking the new cache line as speculative; and
7 allowing the speculative thread to write speculative data to the new cache line.

1 34. (New) An apparatus comprising:

2 a processor including:

3 a first logical processor to execute a speculative thread;

4 a second logical processor to execute a non-speculative thread;

5 a storage area to include a speculation identifier (ID) field, the speculation ID field

6 to hold a first value to indicate an associated store instruction is associated

7 with the speculative thread; and

8 control logic to prevent data associated with the store instruction from being

9 consumed by the non-speculative thread, based on the speculation ID

10 holding the first value.

1 35. (New) The apparatus of claim 34, wherein: the first and the second logical processors are

2 the same logical processor, and wherein the non-speculative thread and the

3 speculative thread are to be time multiplexed for execution on the same logical

4 processor.

1

1 36. (New) The apparatus of claim 34, wherein: the storage area includes a store buffer, and

2 wherein the speculation ID field is included within a store buffer entry of the store

3 buffer, the store buffer entry to also hold a first address associated with the store

4 instruction and the data associated with the store instruction.

1 37. (New) The apparatus of claim 36, wherein: the first value is to include a first identifier
2 (ID) value associated with the first logical processor.

1 38. (New) The apparatus of claim 37, wherein: the control logic includes
2 comparison logic to compare a second address and a second ID value, which are
3 associated with a load instruction that is to be executed as part of the non-
4 speculative thread on the second logical processor, with the first address and the
5 first ID value; and
6 store blocker logic to prevent data associated with the store instruction from being
7 consumed by the load instruction that is to be executed as part of the non-
8 speculative thread, in response to the first ID value and the first address not
9 matching the second ID value and the second address.

1 39. The apparatus of claim 34, wherein: the processor further includes a third logical
2 processor to execute an additional speculative thread, and wherein the control
3 logic is to allow data associated with the store instruction from being consumed
4 by the additional speculative thread.

1 40. (New) The apparatus of claim 34, wherein: the first value is to include a 1-bit mode
2 value.
3 comparison logic.

- 1 41. (New) The apparatus of claim 34, wherein: the processor further includes marking logic
- 2 to set the speculation ID field to the first value in response to detecting the store
- 3 instruction associated with the speculative thread.